**The TKBio workflow as a series of Knowledge Beacon (KB) calls**

# Discover a list of candidate concepts

**Input:** URL encoded string of keywords to match against the canonical or aliases of a concept

**Output:** (paged) list of matching concepts from all KB supporting the API call

**Variants:** Additional keywords and semantic type filters can constrain the list; results are batched as pages

**API endpoint:**

GET /concepts?q=*<keyword1>&*q=*<keyword2>…&*q=*<keywordn>*

# User selects a concept to get at a table of all its relationships

Once a user selects a concept from the list of concepts matched by keyword and semantic group constraints, the intensional meaning of the user’s search is more precise. However, the chosen concept record originated from only one Knowledge Beacon source. In order to retrieve all related data from all available Knowledge Beacons, the workflow much identify all equivalent concepts across those KB’s. To “prime the pump” of the search, the CURIE of the originally matched concept is provided as a list of one element to the /exactmatches API call.

## System identifies equivalent concepts identifiers across KB’s

### Given a list of exact match (e.g. output from II.A.1 above), return equivalent concepts

Concept equivalency is discerned through exact matches (*sensa* **skos:exactMatches** and **owl:sameAs**) of any concept canonical or equivalent identifiers known to the given KB, with at least one of the input CURIEs. Any such “equivalent concept identifiers” not seen in the original input list, are returned as additional identifiers deemed equivalent, to allow iterative discovery of equivalent concepts using this API call.

**Input:** List of CURIES to currently known exact matches concept to the current seed concept of interest

**Output:** List of exact matches (sense **skos:exactMatches** and **owl:sameAs**) associated with the concept

**API endpoint:**

GET /xref?xi=*<xref1>&*xi=*<xref2>…&*xi=*<xrefn>*

### Iterative discover of the equivalent concept clique

The lists of cross-references from II.A.3 above are consolidated into a union set and identifiers that were already initially run are subtracted from the resulting set, then the difference set of identifiers are used in iterative calls to II.A.2 until the resulting “equivalent concept” identifier clique ceases to expand, suggestive of a complete clique (to the extent known by the available KB’s)

## Set of equivalent concepts are used to retrieve related statements.

**Input:** “Equivalence clique” of 1..m concept identifiers (‘cid’ from II.A.3 above)

**Output:** List of matching subject-predicate statements

**API endpoint:**

GET /statements?ci=*<cid1>&*ci=*<cid2>…&*ci=*<cidm>*

# User requests details about a specific concept

**Input:** A specified globally unique user-selected concept identifier

**Output:** A more complete report of properties of the concept

**API endpoint:**

GET /concepts/*{conceptId}*

# User selects a specific statement to get at evidence

**Input:** The evidenceId associated with a given statement (from the output of II.B)

**Output:** A list of citation references supporting the statement

**API endpoint:**

GET /evidence/*{evidenceId}*